

DATABASE Expert

A database designed to help a bird spotter organise his bird-watching activities has left him all a flutter, but luckily Kay Ewbank can lend a helping hand. Plus, linking Access records with PowerPoint presentations



SQL SERVER 2005 IS HERE

Microsoft has announced the product launch dates for SQL Server 2005 and Visual Studio 2005. Both products will be formally launched the week of the 7th November. However, those who wish to try SQL Server 2005 before then can download it. The Community Technology Preview of SQL Server 2005 is available from www.microsoft.com/sql/downloads.

In the same announcement, it was revealed that the first TPC-C benchmark result for Windows Server was over million tpmC, using an early build of SQL Server 2005 running on an HP Integrity Superdome system. The result of 1,082,203 tpmC is 38 per cent higher than that of SQL Server 2000 with over seven per cent higher performance and 35 per cent less cost per transaction than Oracle 10g on an HP Integrity Superdome.

■ DATABASE DEVELOPER AWARDS

If you're proud of a database you've developed, why not enter it for an award? Mark Nellis did exactly that with his iArchitecture database for the construction industry, and he won the FileMaker Business Solution of the Year Award. His team picked up a trophy and £2,000 worth of software at the Macworld awards in May. The runners-up, who each received £500 worth of FileMaker software, were iTelligence Classrooms and Time Harvest software.

Q I am trying to design a database for birds seen on bird-watching trips. I want my database to generate reports automatically for inclusion on our local RSPB website. However, I am having a little difficulty putting the database together.

The database has three main tables: one for trips, one for locations and one for birds seen. I need a way to arrange it so that it starts with a trip, then the user can add locations and birds. My latest attempt had a subform for location and a sub-subform for birds (a subform of location). This threw up a key index error when I tried to enter a second bird at a location.

My three tables for trips, locations and birds (called tblTrip, tblLocation and tblBirds) each have an index field. In addition, I have a couple of linking tables that associate trips with locations and locations with birds. Each record in tblTrip2Location has two lookup fields: one for a trip and one for an associated location. TblLocation2Birds is similar, associating locations with birds that have been seen there. How do I make this work?

John Fairhurst

A It's easy for things to get in a bit of a pickle when a database has five tables, especially if you start nesting subforms. You really should simplify your database.

First let's look at the source of your immediate problems, which is Access's Lookup Wizard. Many Access gurus think the Lookup Wizard causes more trouble than it's worth. You'll find the Lookup Wizard in the table designer in Access. It is meant to guide you through the process of creating a field that gets its data from a list of values you enter when designing the table, or from values already present in another table. This sounds like a good idea, and it can help you ensure the user enters only the data you want them to enter.

If you examine a table with a field that has been defined using the Lookup Wizard, you'll see that the field you defined is shown as having a data type of a number. If you then look at the lookup tab of the field definition, you'll see that the display control is a combo box. The row source type for this combo box is shown as an SQL Select query.

In your database, in the table called Trip2Location that gets a location from the main location table, the lookup field has the following SQL code:

```
SELECT [tblLocation].[Index],
[tblLocation].[Location] FROM
tblLocation;
```

That is, it lets users select any location from the Location table. When the user attempts to enter data into the field, they'll be shown a combo box, even if they're in the datasheet view. When the field is added to a form, it will also be shown as a combo box, ensuring easy and error-free data entry.

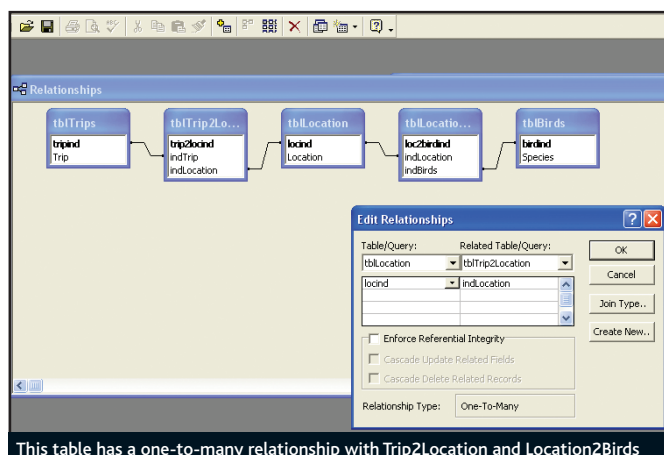
RELATIONSHIP ISSUES

So far, so good, but there are problems with using the Lookup Wizard. One of the most pernicious is that when you create a lookup field in a table, Access creates a hidden relationship between the tables and sets up a hidden index. The relationship type it creates is a one-to-many relationship.

You can see the hidden relationships in your database by choosing Relationships from the Tools menu in Access, then 'Show all' from the Relationships menu. In your database, the Location table has a one-to-many relationship to the Trip2Location table, and it also has a one-to-many relationship to the Location2Birds table. In other words, a record in the Location table can be related to many records in the Trip2Location table, and to many records in the Location2Birds table. However, any record in the Trip2Location table can be related

tblbirds : Table		
	Field Name	Data Type
	birdind	AutoNumber
	birdmainind	Number
	locind	Number
	lookupfield	Text
		Text
		Memo
		Number
		Date/Time
		Currency
		AutoNumber
		Yes/No
		OLE Object
		Hyperlink
		Lookup Wizard...

Use the Lookup Wizard in Access to create lookup fields



This table has a one-to-many relationship with Trip2Location and Location2Birds



to only one record in the Location table, and any record in the Location2Birds table can be related to only one record in the Location table. The screenshot on the previous page shows all the relationships in John's database.

NESTING INSTINCT

The relationships cause problems when combined with the use of nested subforms. In Access subforms, the main form and subform are linked so that the subform displays only records that are related to the current record in the main form.

Your main form shows the records from the Trip table. It has a subform showing the Location2Trip records. But the subform itself has a subform showing the Location2Birds records. So when a record is added to the list of birds in the Location2Birds subform, its link tells it only to accept records related to the record currently displayed in the Location2Trip subform. At the same time, the lookup field has created a relationship not to Location2Trip but to the main Location table.

On the Location2Birds subform, which should be getting its location value from the form it's nested in, the row source is instead:

```
SELECT [tblLocation].[Index],
[tblLocation].[Location] FROM
tblLocation;
```

The subform rule tells Access, 'Show me the records that are matched to the record on the form I live in'. However, the hidden relationships contradict this, saying, 'These records match a different table altogether'. There is a quick fix, though, which is to alter tblLocation2Birds so instead of looking up the location in the location table, it looks up a location from the table trp2Location.

However, the best idea would be to design the database differently. You should redesign the table structures, or at least the form structures. One solution would be to change the forms. You could use two simpler forms; one form to enter the locations visited on a trip and a separate one to enter the birds seen at particular locations.

A better alternative would be to change the table structure. You could add a single Sightings table to your three main tables for Trip, Location and Birds. Each record in the Sightings table would combine data from the Trip, Location and Birds tables, along with a time and date for the sighting.

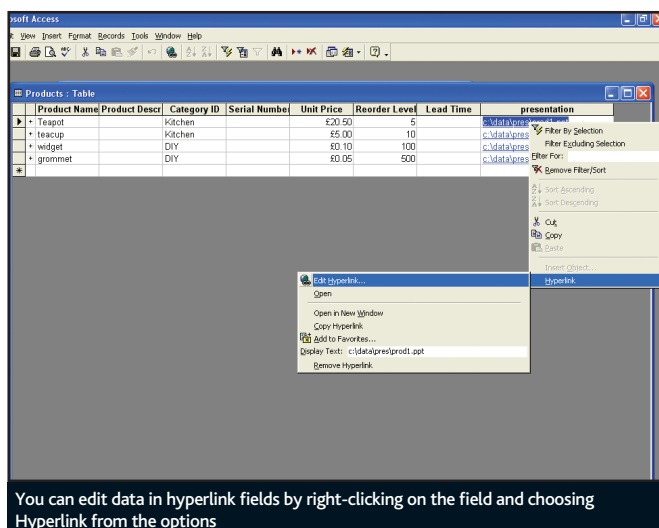


I have a problem with an Access 2000 database. I have a table with several hundred records in it, and I would like to link each record to its own individual PowerPoint show, so it opens when you click on it. I know that I can use the hyperlink field and create each link individually, but is there any way to automate the process, perhaps using a macro? Also, is there an alternative way of linking records from Access to an external file?

John Richards



If your PowerPoint presentations have names that can be generated automatically, you can automate the process. Suppose we wanted to add presentations to the Products table in the *Computer Shopper* database. The first step would involve adding a field of type hyperlink to the table. We could then enter



hyperlinks manually by right-clicking on the field and choosing Hyperlink from the options offered. This shows a further menu from which you can choose the 'Edit the Hyperlink' option. You can then type the address or browse to the correct location.

A complication with hyperlinks is the fact that they have several elements. The two main elements are the display name (what you see onscreen) and the actual hyperlink address (where you go when you click).

PRODUCT PLACEMENT

Automating the process of making hyperlinks is a little tricky. The following code works for the database called Shopper, with a table called Products. Products has a field called ProductID that holds the product ID, and another called Presentation that is the Hyperlink field.

I'm assuming that all the presentations are called prodxxx, where xxx is the product ID. So the product with ProductID 3 would have a presentation called prod3. I'm also assuming that the presentations are stored in the folder c:\data\pres, so the file path and name for the presentation for the product with ProductID 3 would be C:\data\pres\prod3.ppt:

```
Private Sub Command0_Click()
Dim db As Database
Dim rsProducts As Recordset
Set dbs = OpenDatabase("Shopper")
Set rsProducts = dbs.OpenRecordset
("Products")
With rsProducts
.MoveFirst
While Not .EOF
.Edit
thispres = "c:\data\pres\prod"&
![ProductID] & ".ppt"
thispres = thispres & "#" &
thispres
![Presentation] = thispres
.Update
.MoveNext
Wend
End With
End Sub
```

To run this code, you would need to put a command button on a form and insert the code into its OnClick event. How the code works is

obvious in places, but more obscure in others.

The outer part of the code is fairly obvious. A recordset called rsProducts is used to work through the data from the Products table, looking at each record in turn. Consider the line:

```
With rsProducts
```

This means that in successive lines we can just refer to .MoveFirst and !Presentation, rather than:

```
rsProducts.MoveFirst
```

and:

```
rsProducts!Presentation
```

The filename of the PowerPoint file is built by the line that reads:

```
thispres = "c:\data\pres\prod" &
![ProductID] & ".ppt"
```

This puts together the name from the file path, the product ID and the file extension of .ppt. The only obscure part of the code is the line that reads:

```
thispres = thispres & "#" & thispres
```

Hyperlinks with the two elements mentioned above are stored in Access as a single text string, with the name that you see in the Hyperlink field, followed by a hash symbol, followed by the actual hyperlink address. We have the actual address stored in thispres. This combination of the address, followed by the hash symbol, followed by the address again means the real address is displayed as well as linked to. You could choose something different for the display element of the hyperlink if you wanted.

I was using PowerPoint files stored locally on the C: drive. You could refer to files stored elsewhere by using a URL for a website address, or a universal naming convention (UNC) address.

UNIVERSAL APPEAL

UNC provides a machine-independent means of locating the file. A UNC name usually comprises a reference to a file in a shared folder on a PC accessible over a network. For example, the UNC name for a database named Northwind.mdb in a shared directory named Samples on a computer called MyWorkstation would be \\MyWorkstation\Samples\Northwind.mdb. The UNC is particularly useful in web-based applications, because it allows you to reference data that is not necessarily stored on a particular web server. **CS**

CONTACT

KAY EW BANK

Kay is *Computer Shopper's* database expert and has worked as a consultant and author for 20 years

kay@computershopper.co.uk